Sebelum kita masuk ke pembahasan git, kita akan membahas terlebih dahulu tentang version control. **Apa itu version control** : Version control adalah system yang merecord/ mentrack perubahan file atau sekumpulan file dari waktu ke waktu, jadi kita dapat melihat file dikemudian hari berdasarakan versi yang terrecord di version control.

Jika kita seorang developer/programmer yang membanguna aplikasi, misal aplikasi A dengan versi 1.0 untuk release pertama dan pada waktu berikut nya aplikasi ada pengembangan sehingga naik versi menjadi 2.0. Untuk menjaga/mentrack atau membedakan source code dari versi 1.0 dengan versi 2.0 maka kita membutuhkan system yang menjaga atau mentrack perubahan nama system tersebut adalah VERSION CONTROL. Beberapa hal umum, yang dapat dilakukan oleh system version control antara lain

1. Mengembalikan/merevert file ke versi sebelumnya
2. Mengembalikan project kembali ke versi sebelumnya
3. Membandingkan perubahan dari waktu ke waktu
4. Melihat siapa yang melakukan perubahan dan waktu perubahan pada file tertentu

Dan masih banyak lagi, jadi dengan menggunakan system version control kita dimudahkan untuk berkolaborasi dalam membuat projek yang pengerjaannya dilakukan secara tim.

Macam-Macam Version Control

1. **Local Version Control System**

Sebelum adanya system version control, metode tradisional yang dilakukan kebanyakan orang untuk membedakan versi satu dengan versi lainnya yaitu dengan mengcopy file ke folder atau kedalam direktory lainnya. Dengan metode seperti diatas akan banyak kesulitan yang ditemukan seperti membandingakn perubahan file dengan versi sebelumnya, atau terkadang kita lupa folder mana yang menjadi versi terakhir dari perubahan file. Dengan adanya masalah tersebut programmer mendevelop VCS yang mempunyai penyimpanan sederhana yang menjaga perubahan file dibawah system version control, berikut gambaran nya.

1. **Centralized Version Control System**

Permasalah yang muncul dari local version control antara lain, kesulitannya kita dalam berkolaborasi dengan orang lain. Sehingga untuk memecahkan masalah ini muncul lah CVCS Centralized Version Control Systems. Macam-macam CVCSs yang sering digunakan antara lain CVS, Subversion Perforce dan lain-lain. Dengan CVCSs file atau projek disimpan disatu server dan tim yang butuh file atau projek tersebut perlu checkout dari sentral atau pusat. Dalam beberapa bertahun-tahun CVCSs adalah standar untuk version control. Berikut gambaran dari CVCSs

Setup version control dengan model CVCSs terdapat beberapa keuntungan antaralain, setiap orang dapat mengetahui sampai berapa jauh kontribusi orang dalam projek tertentu, Administrator memiliki kendali penuh terhadap siapa saja dan apa saja yang dapat dilakukan. Dan jauh lebih mudah untuk mengelola CVCSs(version control sentralized) dari pada menangani version control pada setiap lokal client.

Namun setup dengan metode CVCSs juga memiliki kekurangan, ketika server down selama satu jam dalam waktu satu jam pula tim tidak dapat berkolaborasi atau menyimpan versi perubahan file. Kekurangan lainnya antaralain , jika sentral penyimpanan rusak dan backup file belum disimpan maka kita kehilangan segalanya – seluruh sejarah projek atau track projek hilang kecuali snapshot yang kebetulan dimiliki oleh setiap tim dilokal komputer mereka.

1. **Distributed Version Control System**

Metode version control DVCS, local computer tidak hanya memiliki salinan terbaru dari repository pusat, melainkan mereka sepenuhnya mencerminkan repository termasuk riwayat lengkap dari perubahan file. Jadi ketika server down maka repository klien dapat disalin ke repository server untuk memulihkan repository server. Berikut gambaran lengkap dari DVCS

Jadi dengan version control dengan metode DVCS ini projek memiliki beberapa salinan tidak hanya di server melainkan ada slainan di klien lengkap dengan versi dan history dari projek yang ada di server. Berikut contoh dari DVCS : GIT, Mercurial, Bazaar or Darcs.

GIT VERSION CONTROL

Git adalah software yang digunakan untuk menyimpan, memanage banyak file(source code), biasanya digunakan dalam pekerjaan yang pengerjaan suatu projeknya dilakukan secara tim. Sebelum menjalankan perintah-perintah yang sering digunakan didalam git, maka pembaca diwajibkan sudah menginstall di laptop atau di pc masing-masing.

Beberapa point yang akan dilakukan di pembahasan kali ini, sehingga pembaca pada akhir pembelajaran dapat paham dari version control dan bisa memanage projek nya tidak hanya di local repository melainkan bisa memanage di hosting service git seperti GITHUB. Berikut point-point yang akan dibahas secara rinci.



1. Memahami workflow Git

Git dalam workflow dibagi menjadi 3 state yaitu working directory(folder), staging area dan git directory atau repository. Ketika kita merubah file kita berada di working directory, ketika kita memilih file yang mengalami perubahan untuk di commit ke repository kita menambahkan file tersebut ke staging area, dan ketika sudah memasuki staging area file yang mengalami perubahan akan kita commit/save ke repository. Berikut workflow dari git



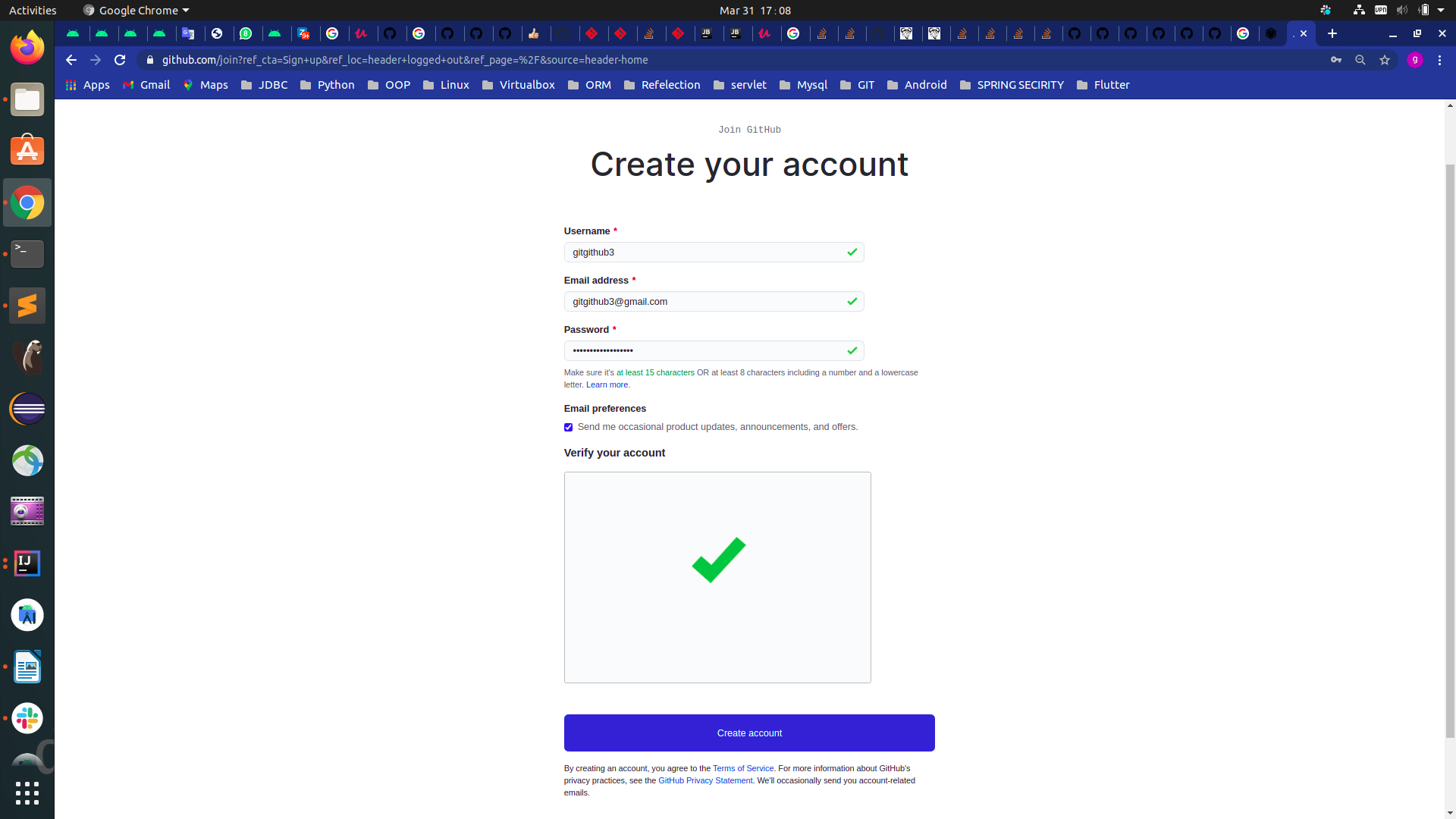
1. Push Exisiting Project git ke Github

* **Buat account github**

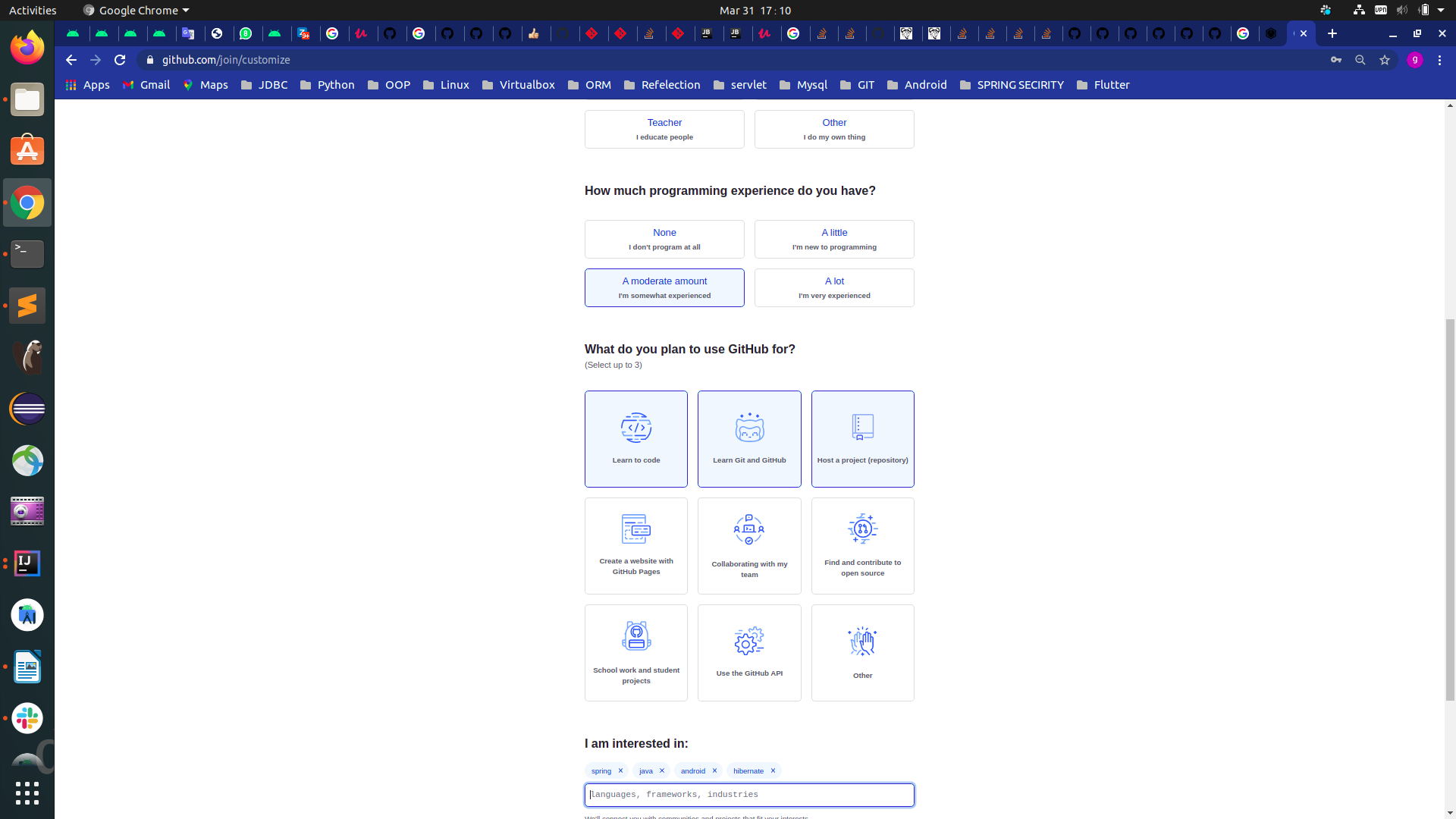
Langkah Pertama yang kita lakukan untuk dapat menyimpan projek kita di github adalah dengan kita melakukan registrasi untuk membuat account github.

**Note : Catat username, password dan email untuk config git dilangkah selanjutnya nanti.**

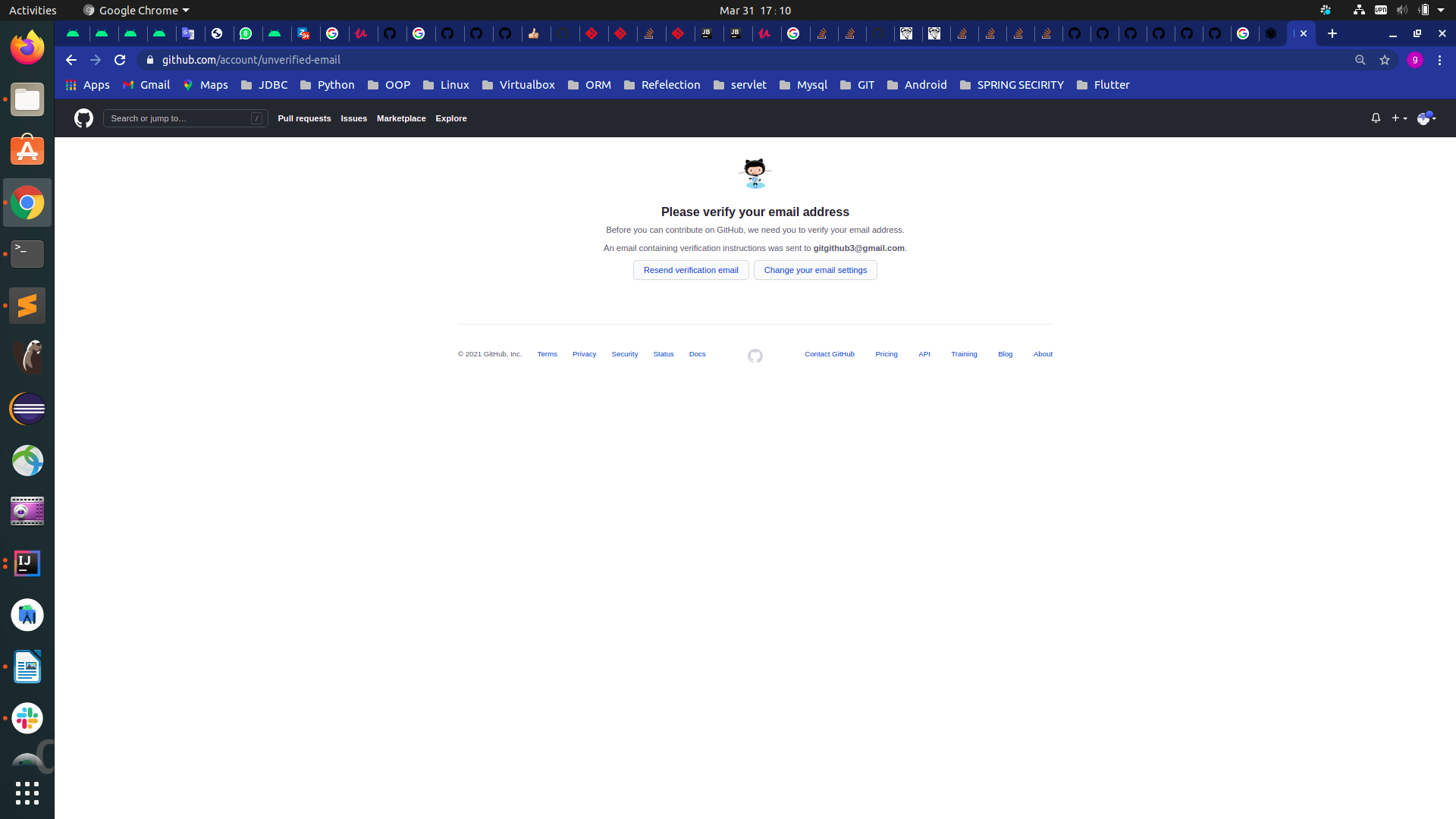
* Masukkan Username, Email dll



* Pilih Aditional info



* Kemudian muncul halaman perintah verifikasi email

Masuk ke email, dan verifikasi. Setelah verifikasi selasai maka proses pembuatan account github selesai.

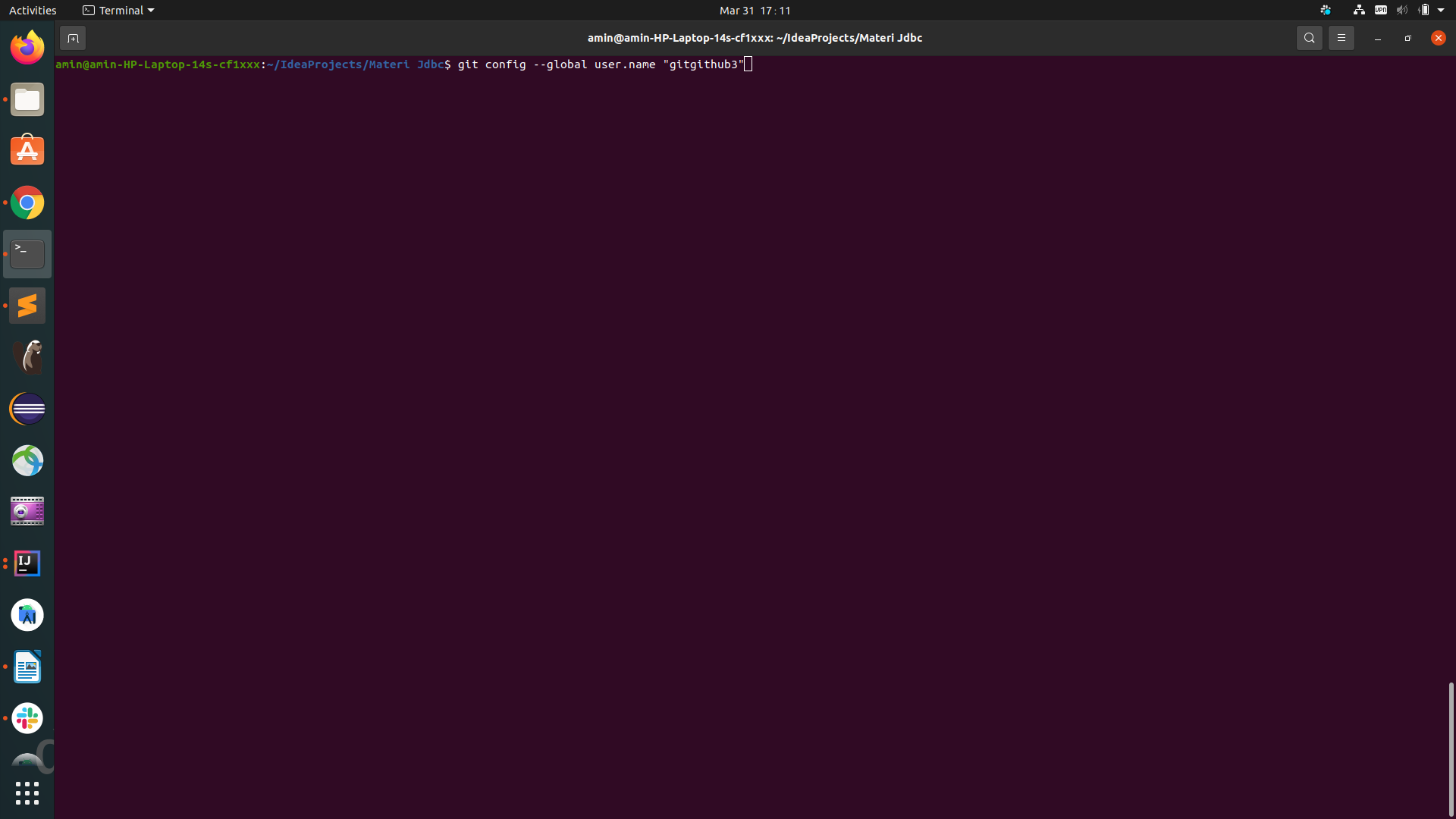
* **Config Username dan Email**

Setelah proses pembuatan account selesai, langkah selanjutnya adalah config username dan email di laptop/computer kita. Sebelum menconfig pembaca diharapkan sudah install git di computer/laptop masing-masing.

Konfigurasi username dan email ini bertujuan untuk mengaitkan username dan email (identitas)anda ketika anda melakukan commit ke local repository atau melakukan push ke github. Jadi setiap perubahan yang dilakukan oleh kita **akan tercatat historynya dengan username dan email yang kita config**. Untuk melakukan config ikuti langkah berikut ini.

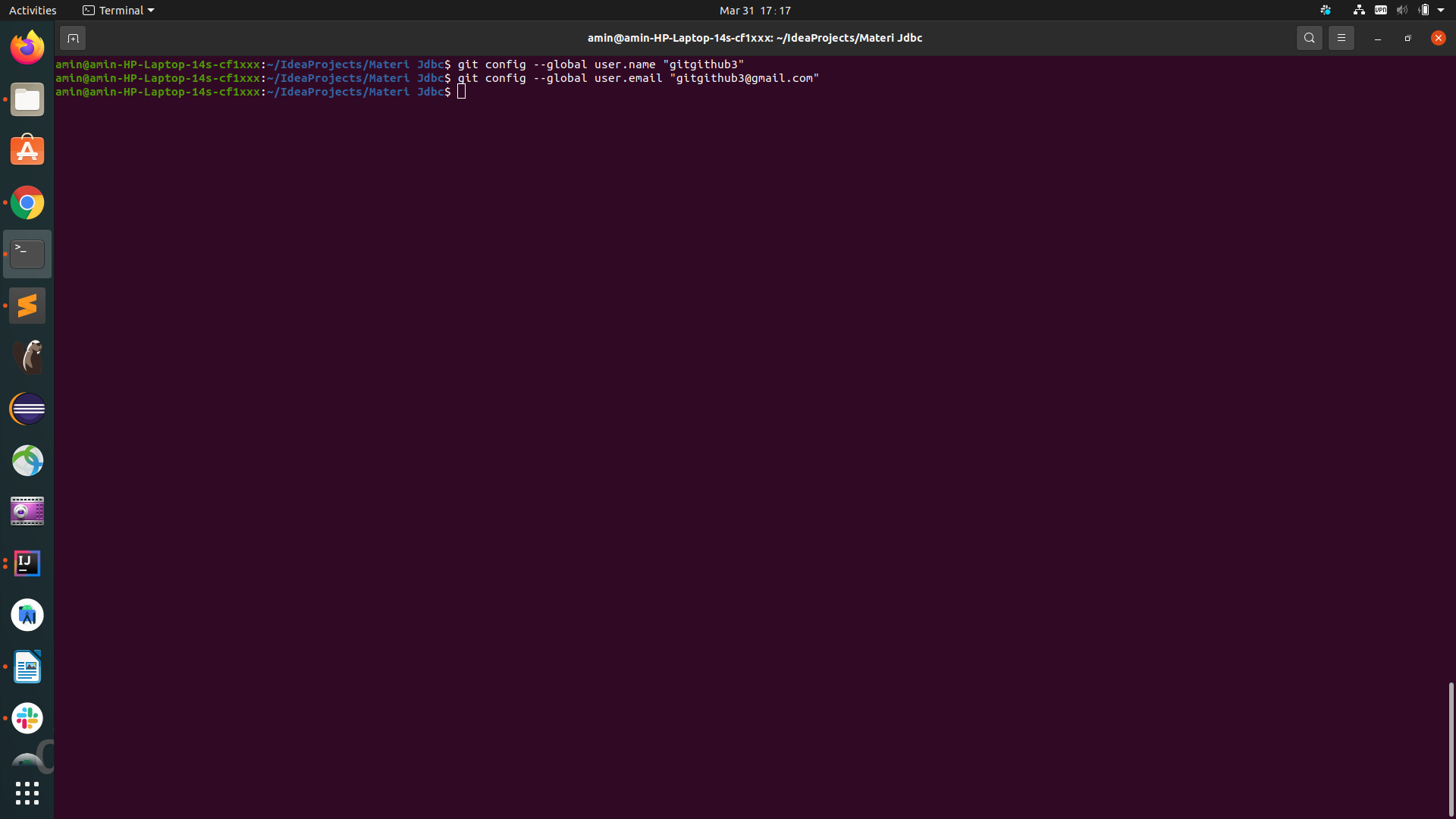
* + **Buka comman prompt/terminal(pastikan git sudah terinstall)**
  + **Ketikkan perintah berikut untuk config username**

Pastikan username git sama dengan nama username github pada saat pembuatan account. (**git config --global user.name "username yang terdaftar di github"**)**.**

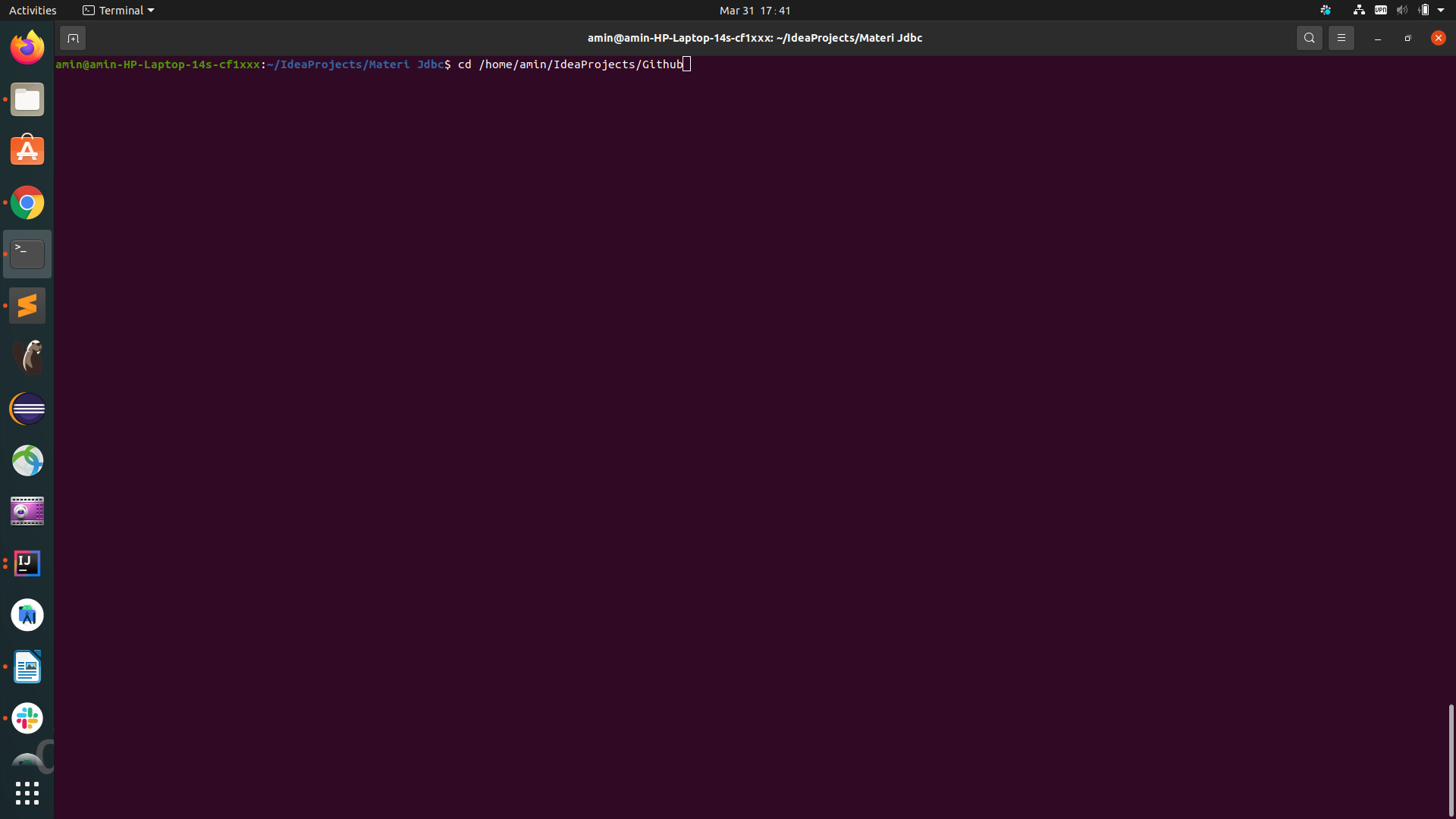


* + **Kemudian config email**

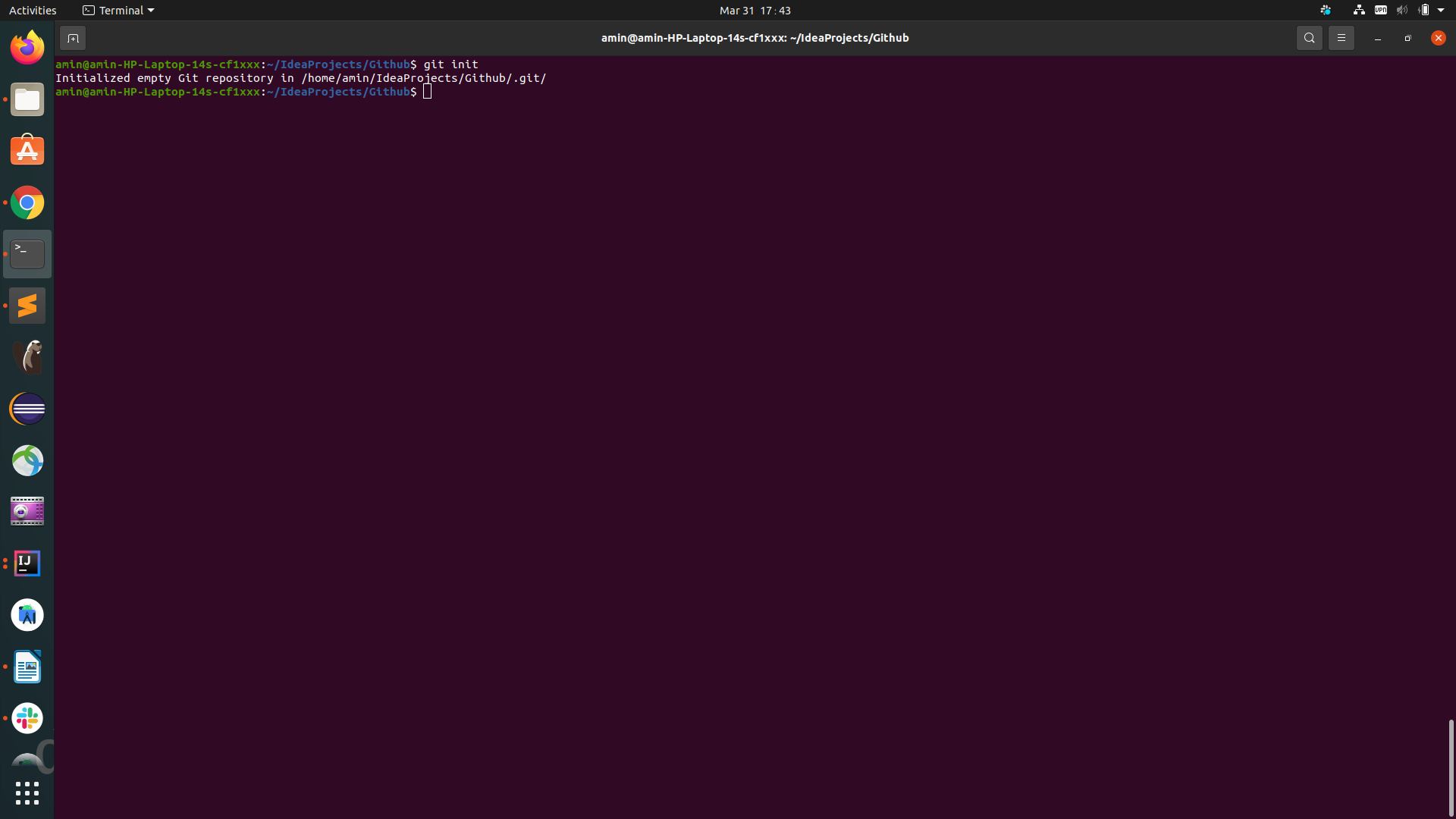
Pastikan email git sama dengan email pada saat pembuatan account github. (**git config --global user.email "email pada saat daftar github"**)



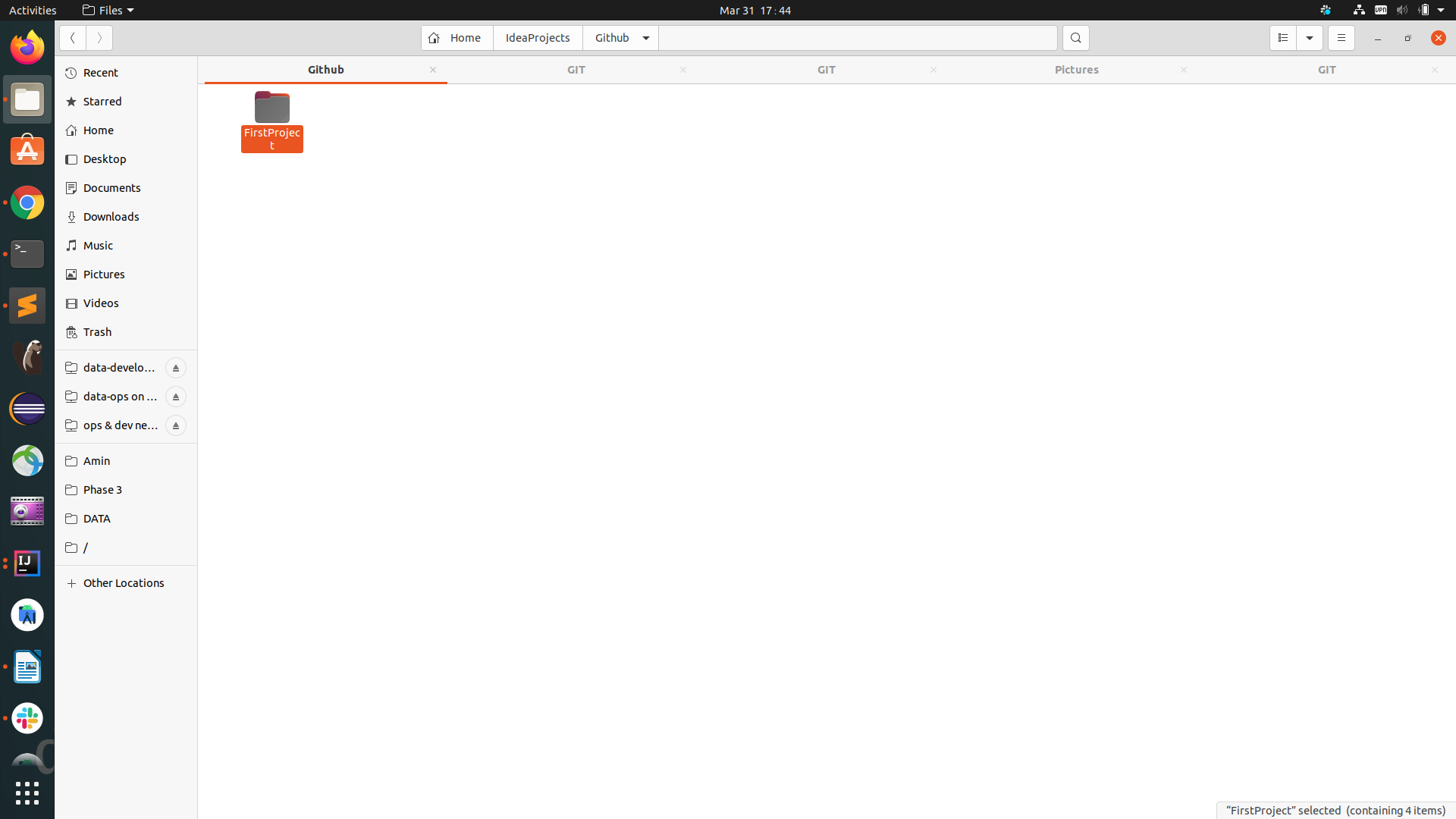
* **Buat Repository local dan commit ke local repository**
* **Tentukan lokasi projek anda**
* **Pindah ke folder dimana anda akan menempatkan projek anda dengan mengetikan seperti pada gambar dibawah**



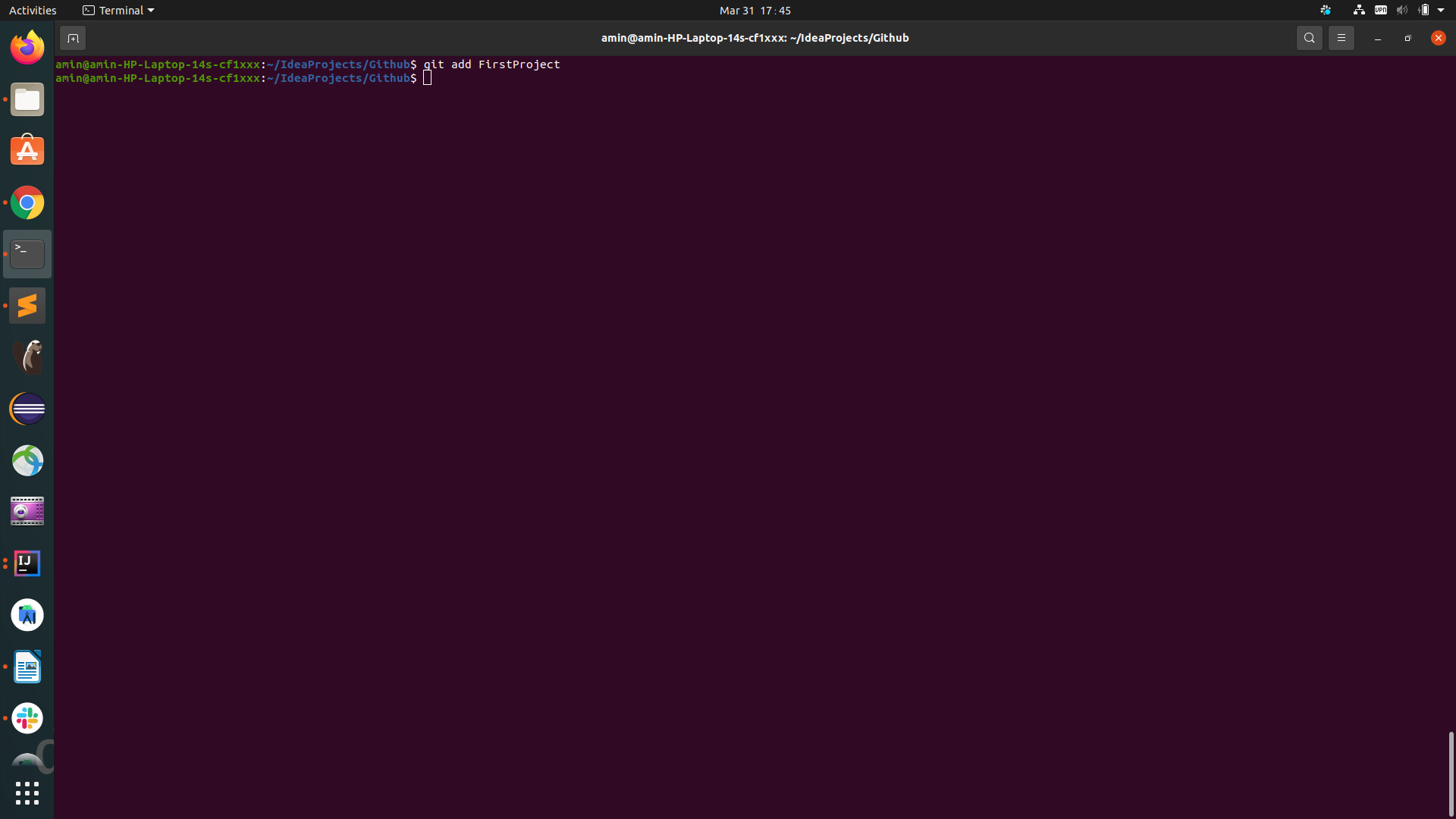
* **Setelah kita pindah ke directory tempat project kita, kemudian kita ketikkan perintah untuk menginisialisasi git (git init)**



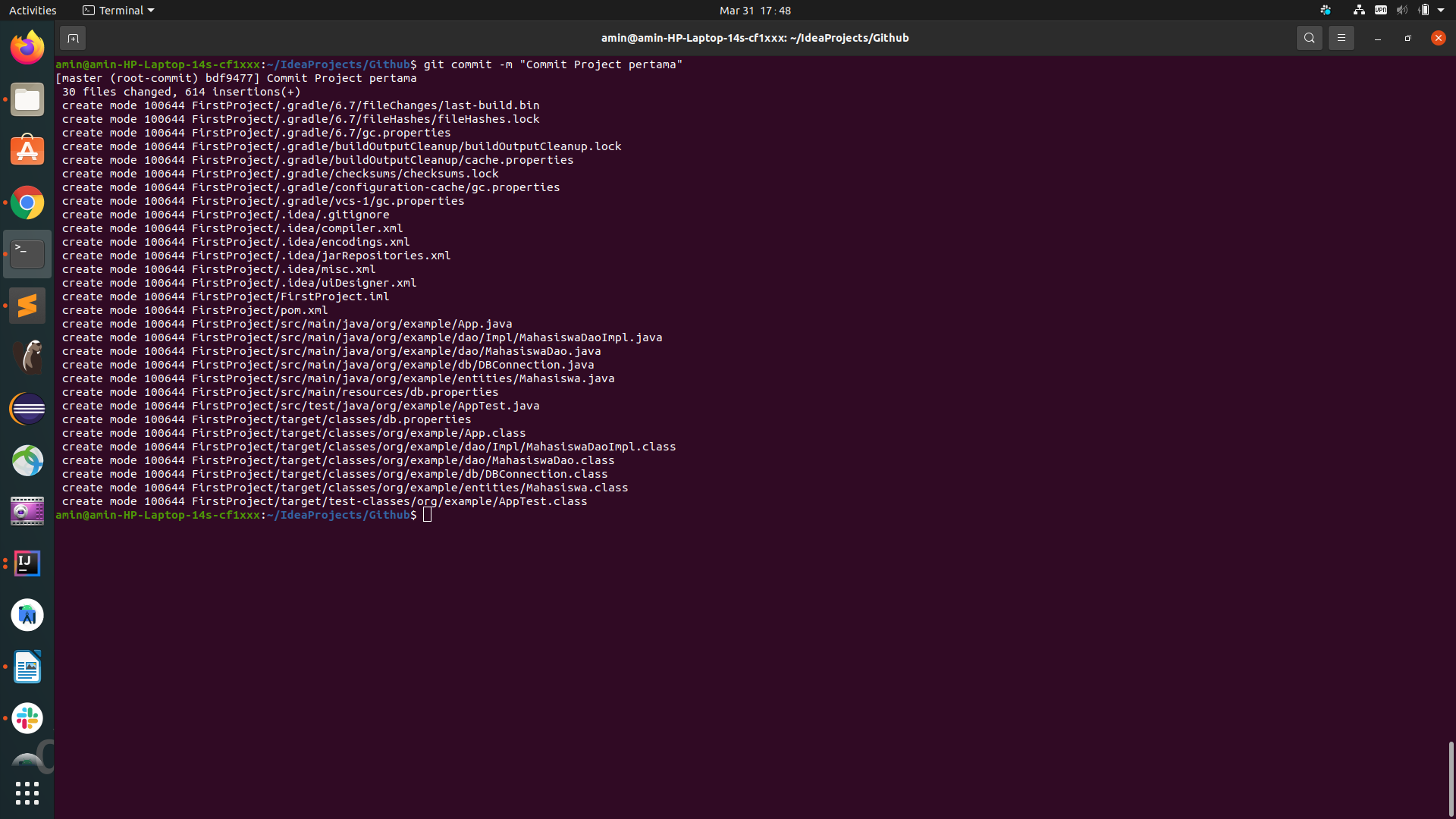
* **Setelah Repository terbentuk di directory tempat project kita berada, langkah selnajutnya Copy Project ke Repository yang baru**



* **Ketikkan perintah untuk menambahkan project ke git**

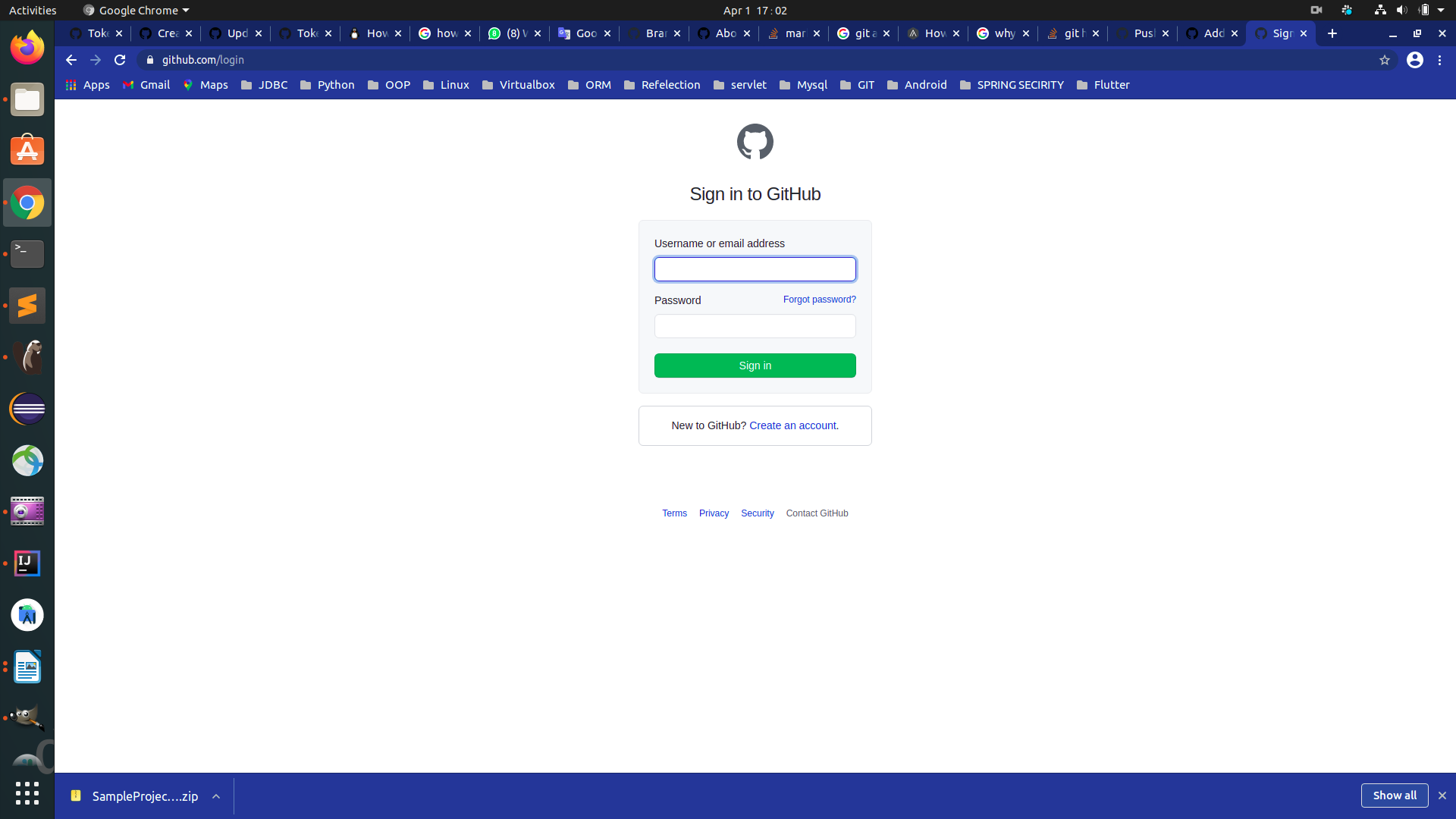


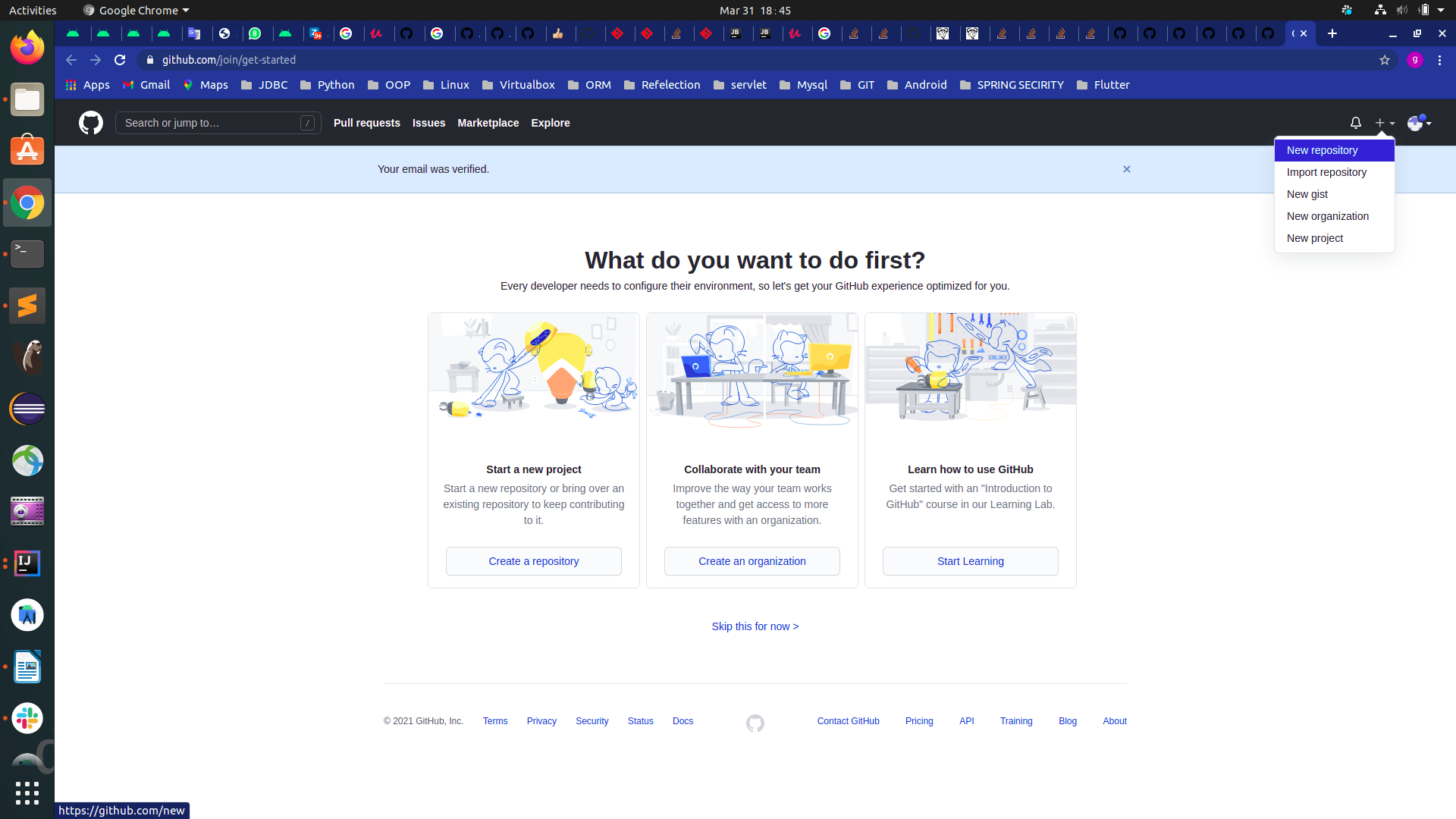
* **Kemudian commit project yang tadi sudah ditambahkan untuk di simpan ke local Repository**



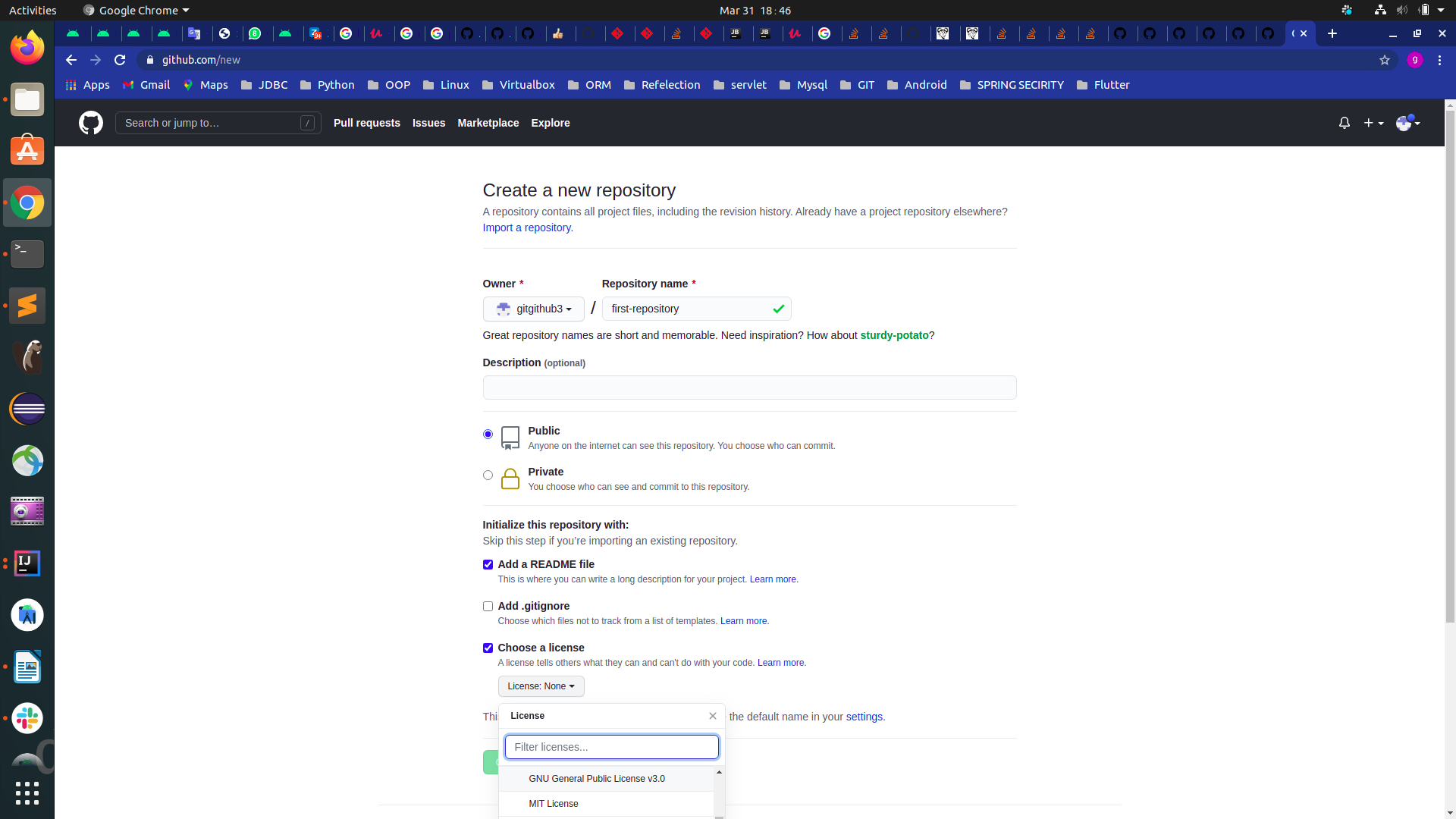
* **Buat repository di github**

Setelah kita membuat account github dan commit project ke local repository, langkah selanjutnya adalah membuat github repository

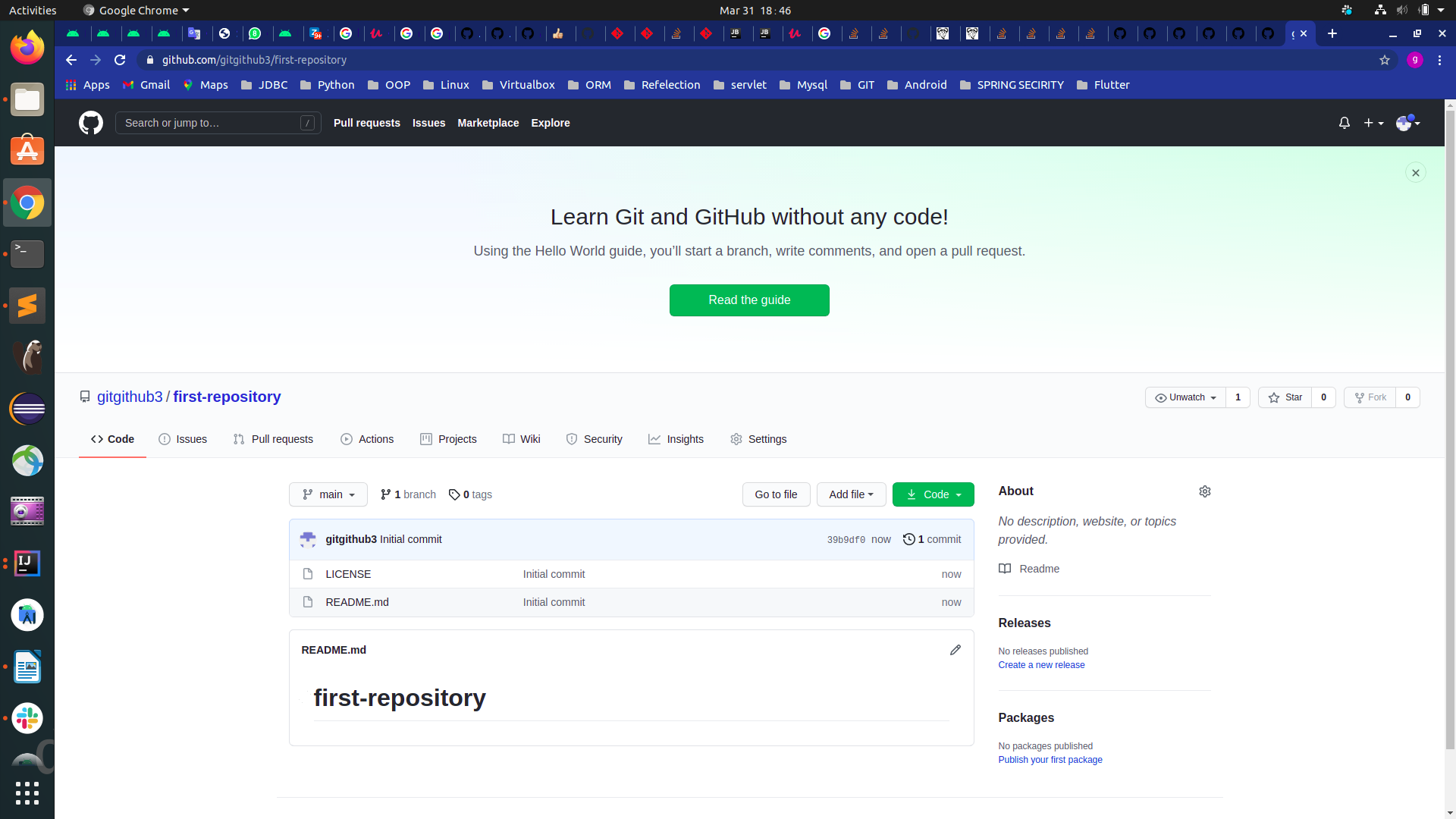
* **Login ke github**
* **Pilih new repository di pojok kanan**



* **Masukkan nama repository, ikuti config yang lain seperti pada gambar**



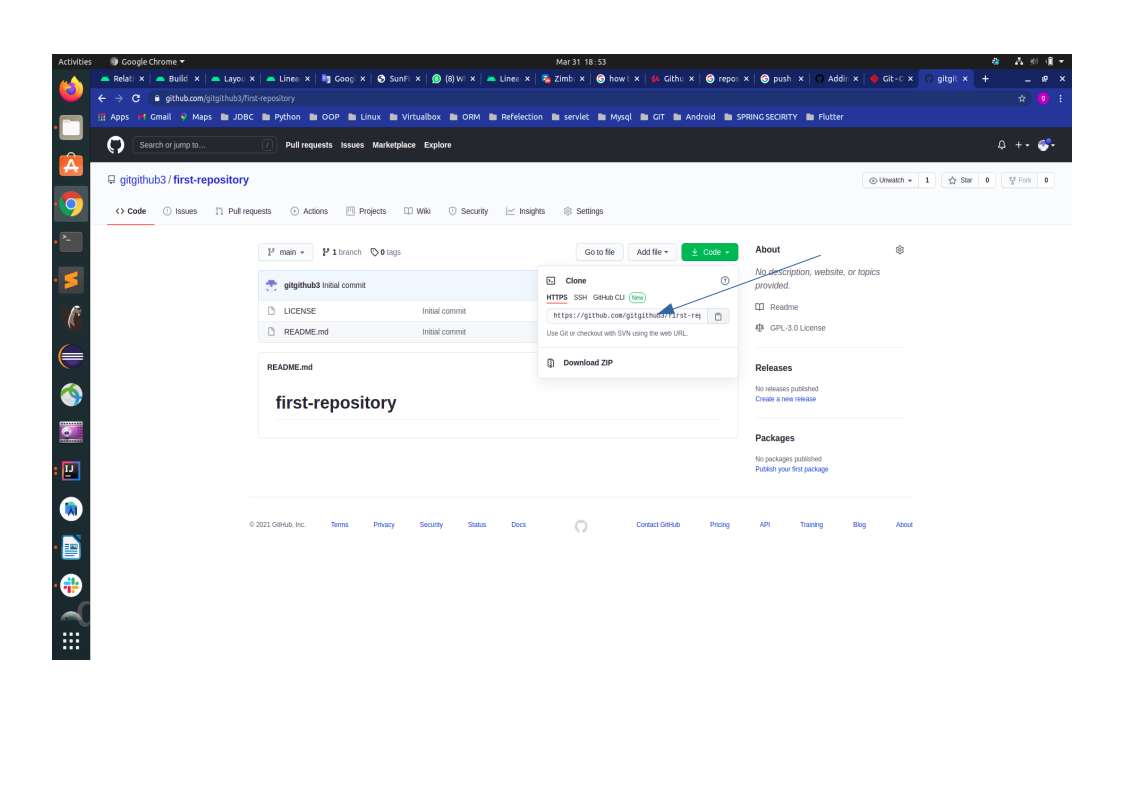
* **Halaman ketika repository sukses dibuat**



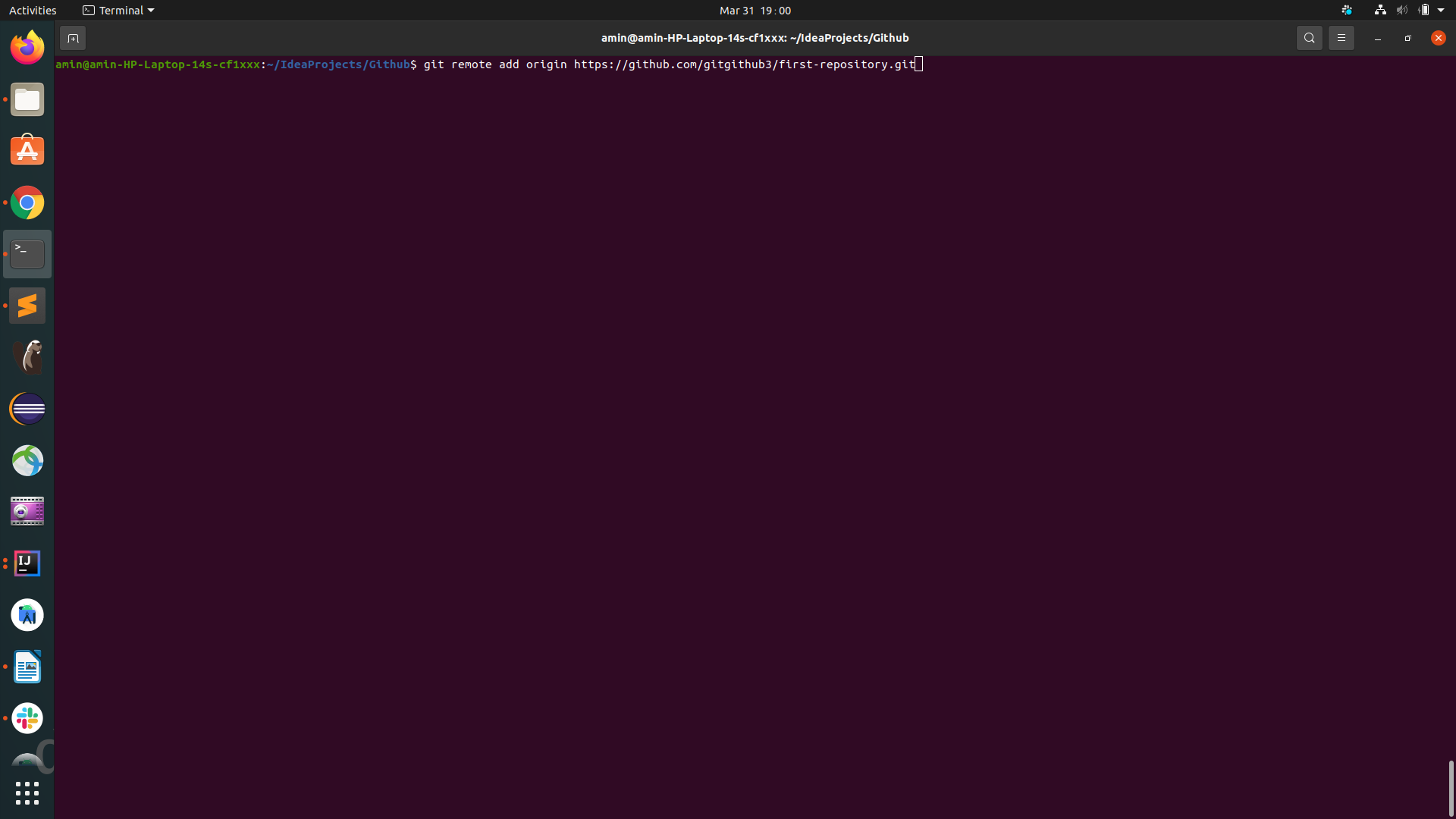
* **Push project ke repository github**

Distep sebelumnya kita sudah berhasil commit projek ke repository local git dan membuat repository digithub. Untuk langkah selanjutnya kita akan push project yang di commit di local repository dan melakukan push ke gihub. Untuk push project, ikuti langkah berikut ini

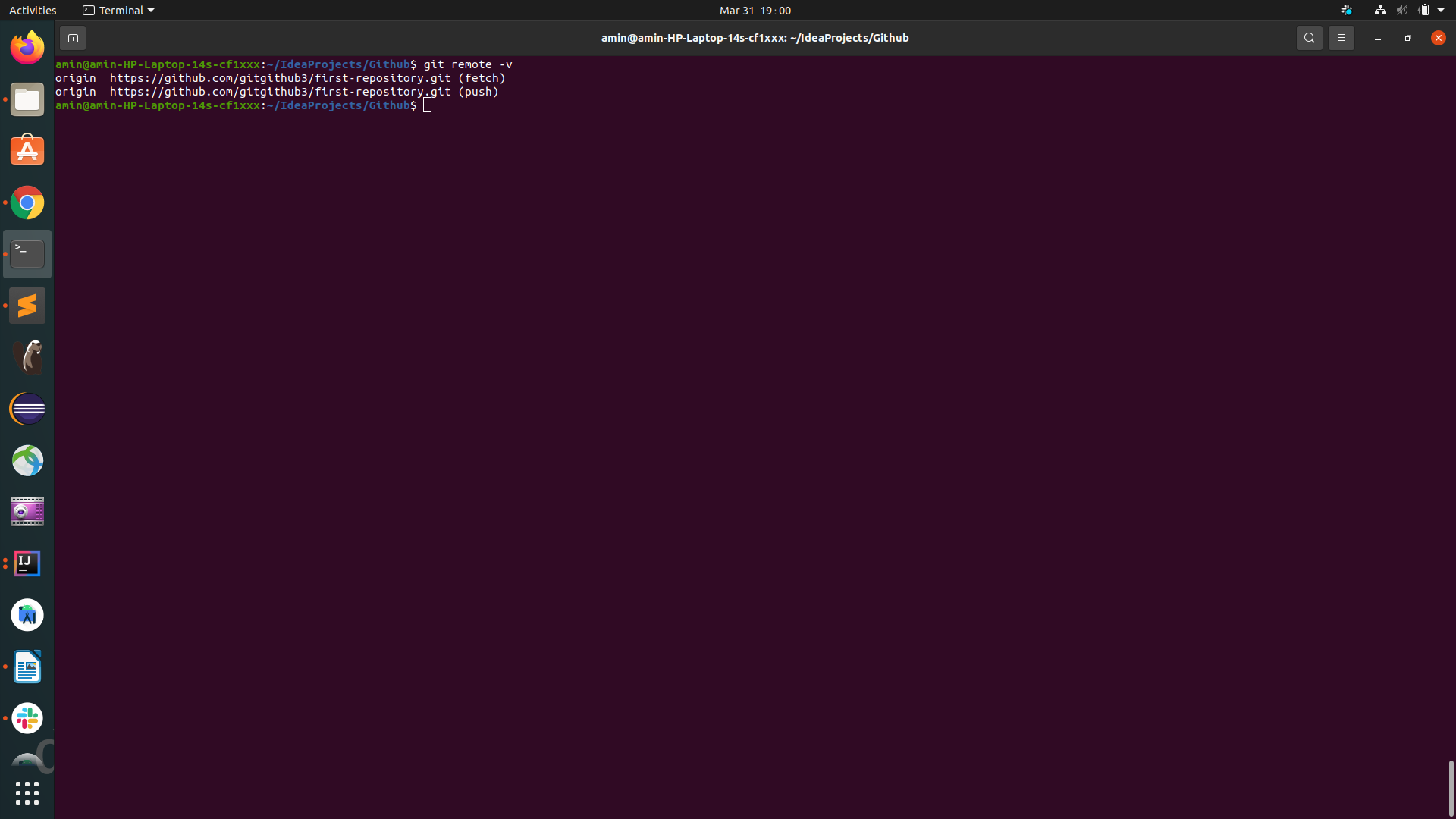
* **Copy URL repository github**



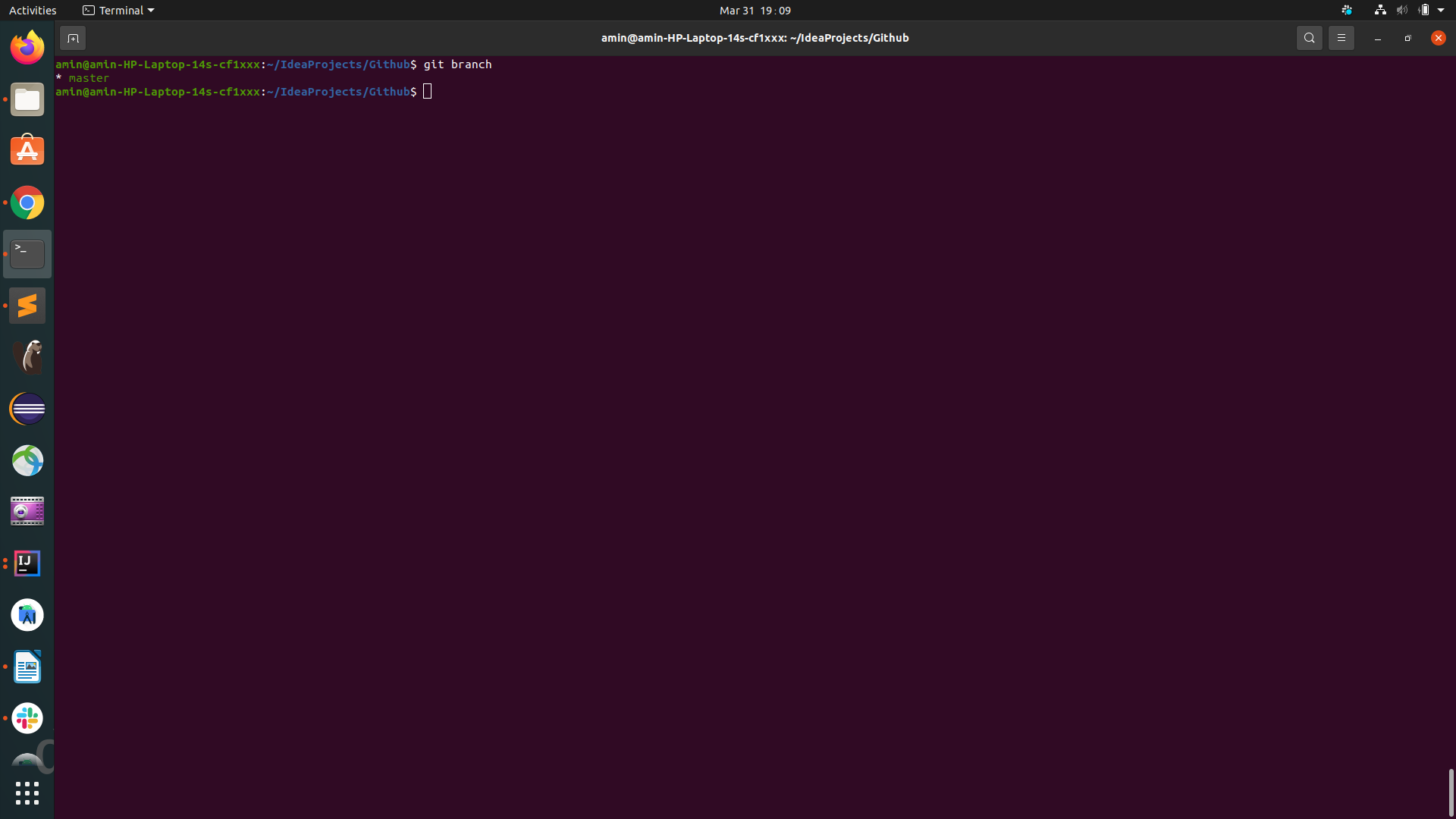
* **Tambahkan URL repository github yang sudah kita copy tadi**



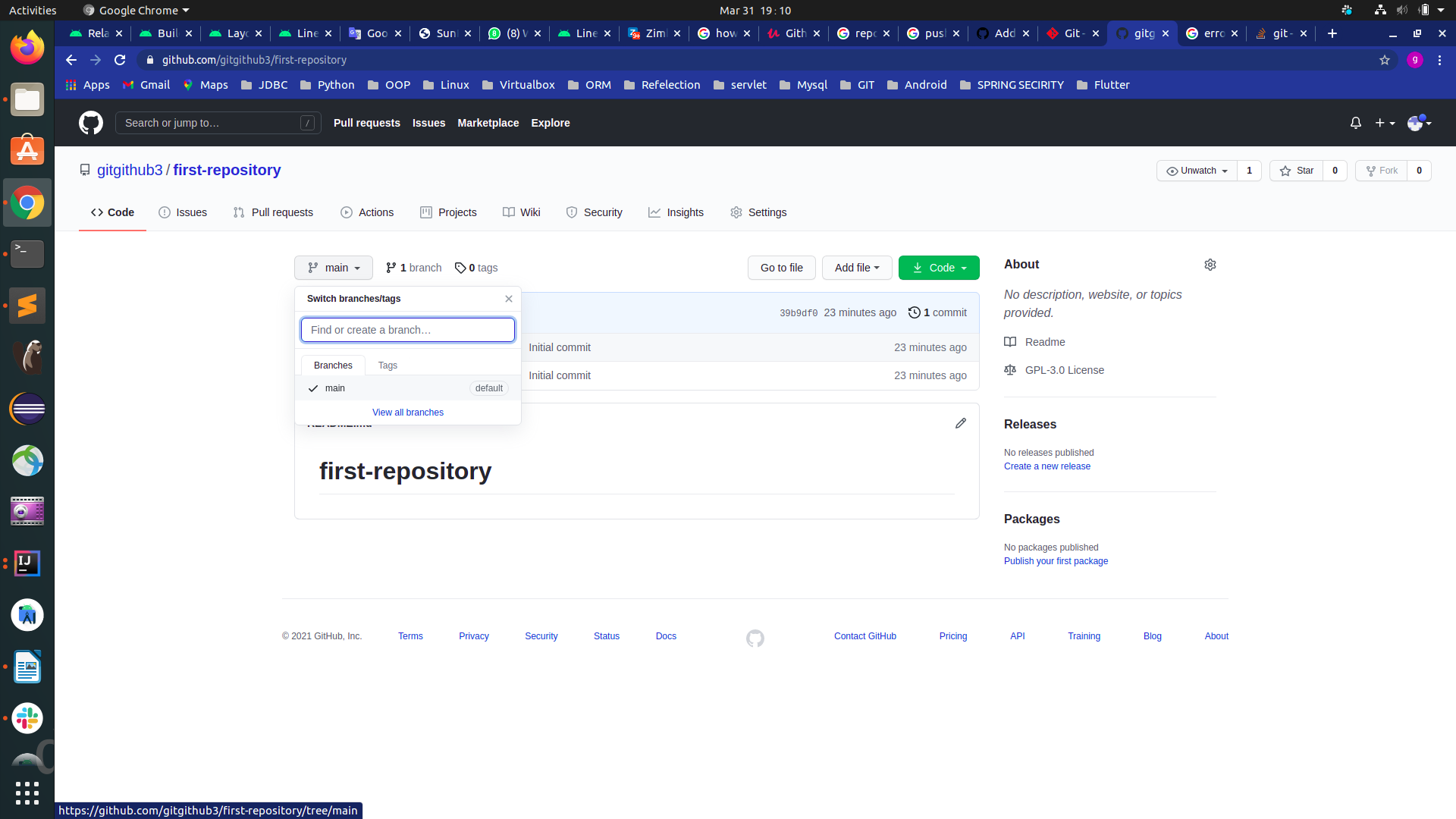
* **Pastikan repository yang di tambahkan sudah berhasil check dengan** perintah **git remote -v**

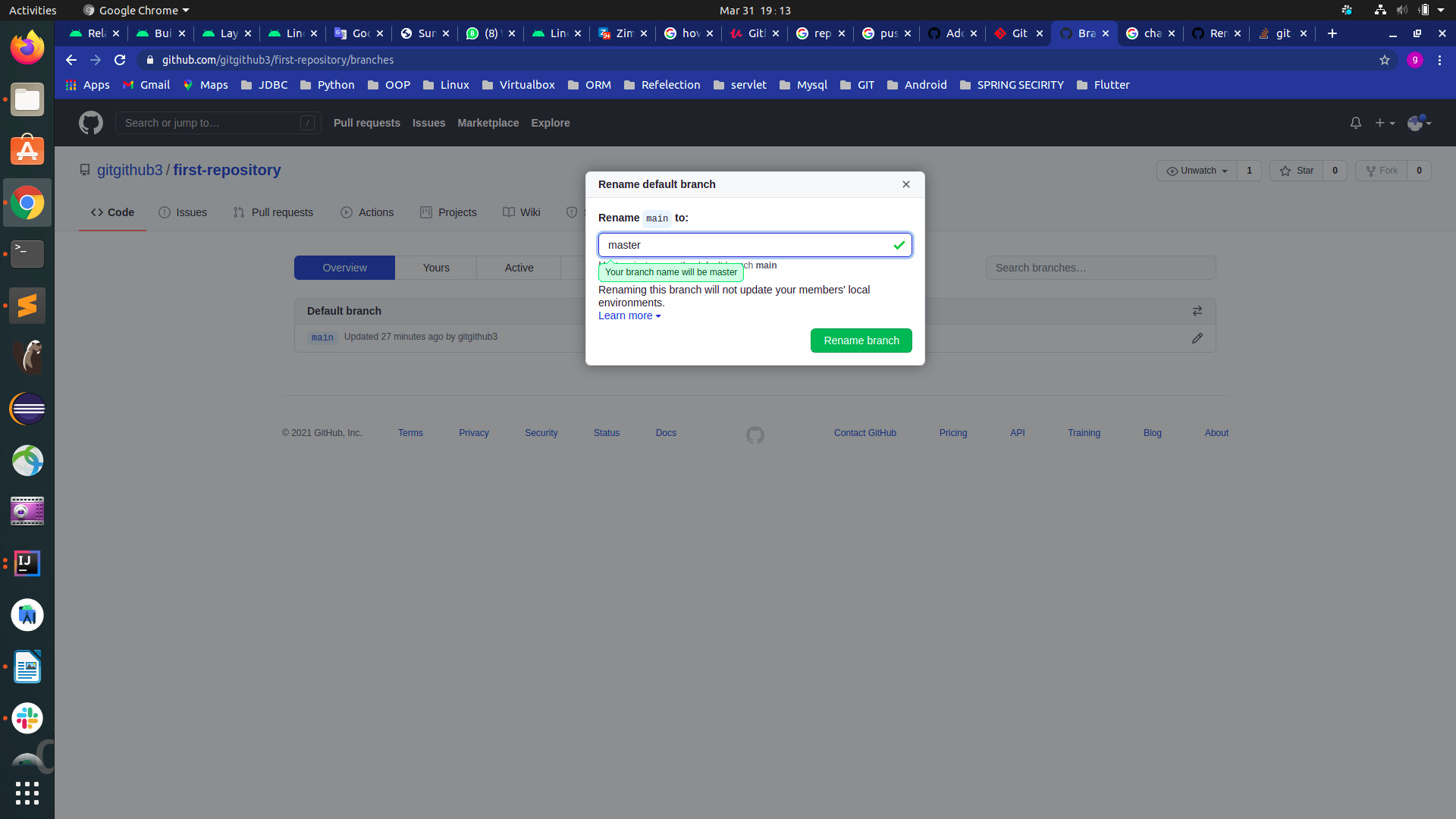
****

* **Pastikan nama branch local dengan nama branch github sama, di git local nama branch nya master**

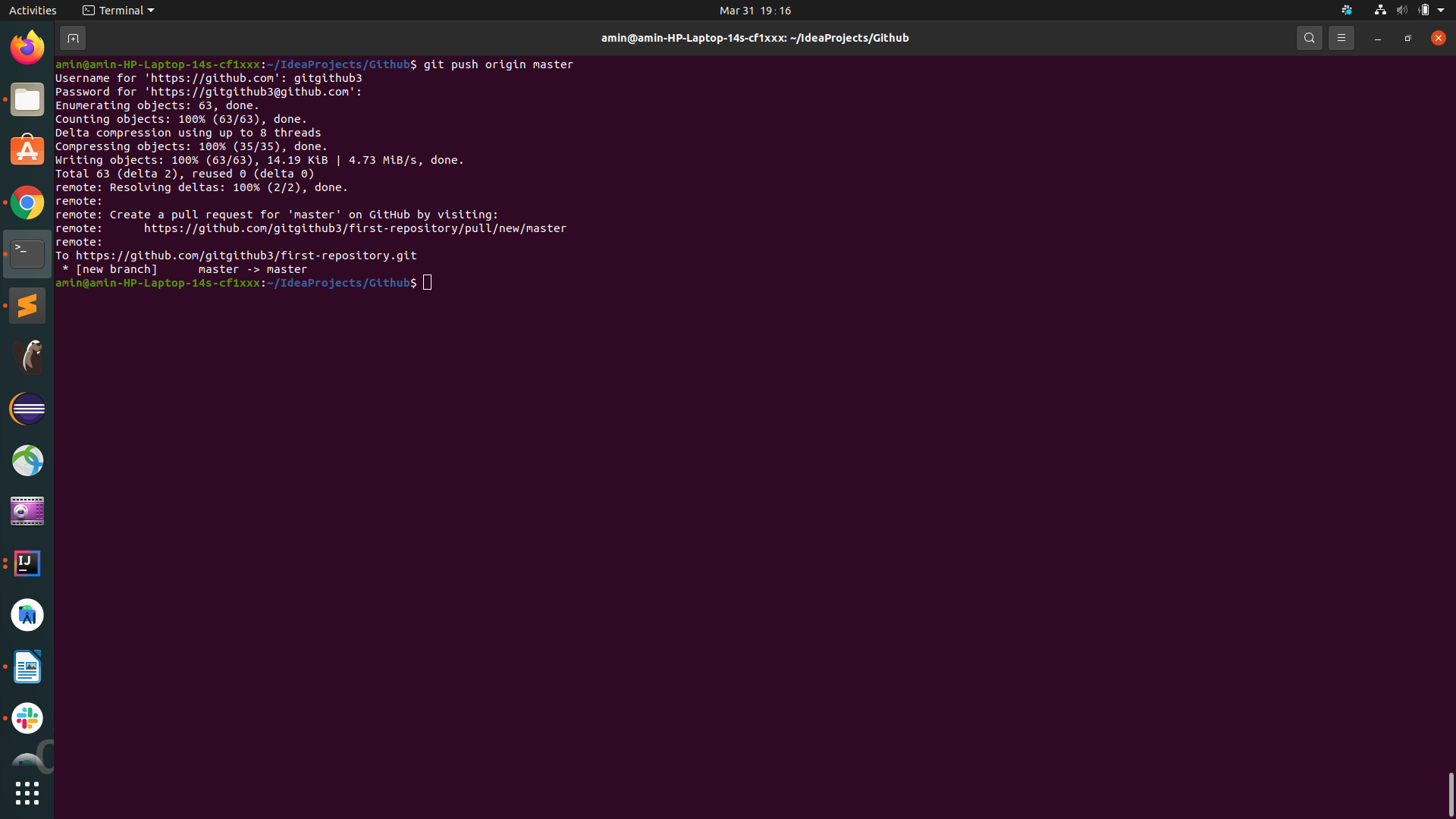


digithub nama branch nya main, perhatikan gambar dibawah

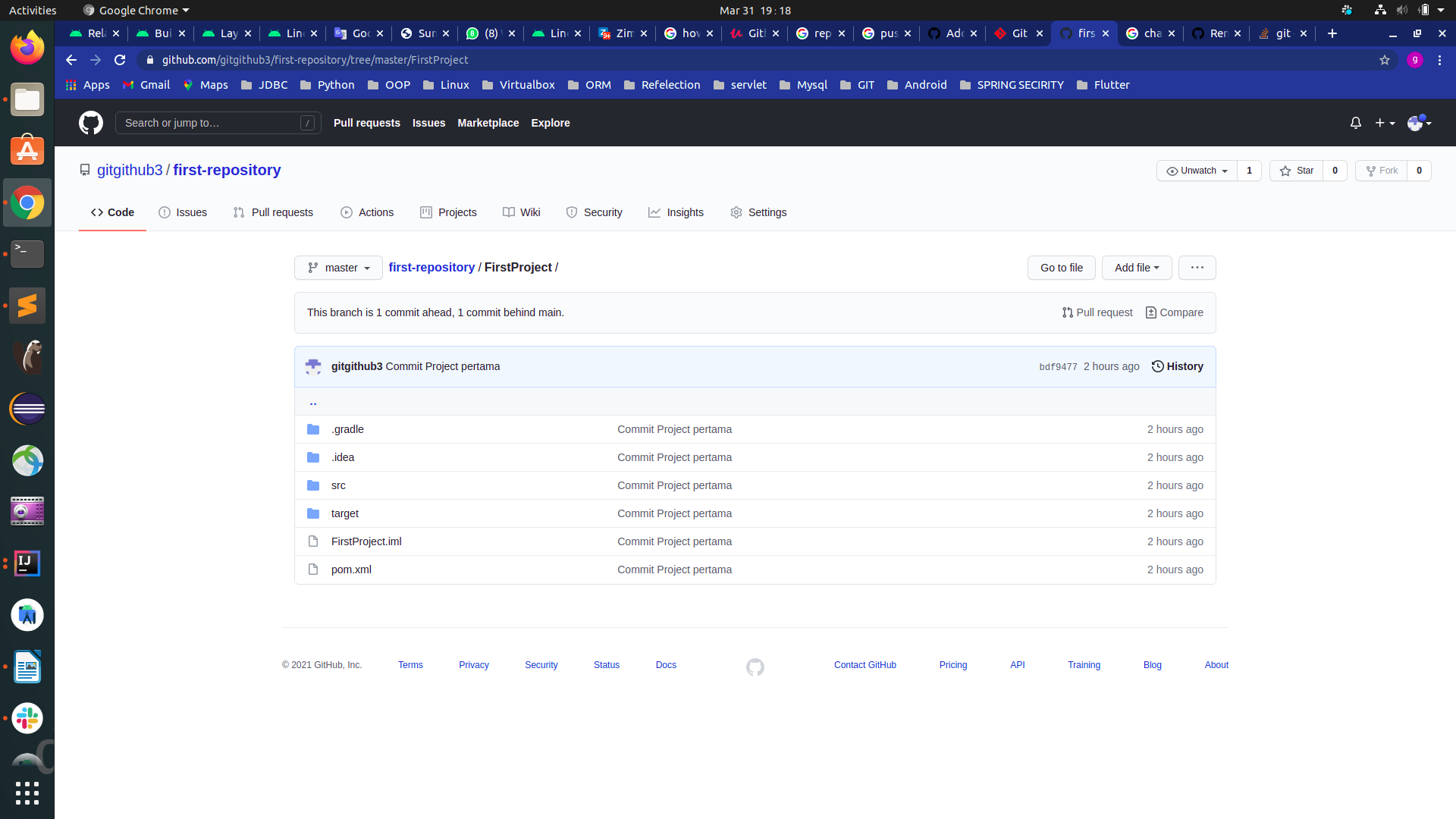
ubah nama branch github menjadi master dengan mengklik tab setting → pilih branch → kemudian klik tanda pensil → dan ubah menjadi branch master seperti pada gambar dibawah



* **Push local repository ke github(pastikan nama git branch nya sama dengan github)**

Masukkan username, dan password (pada saat kita daftar di github)

* **Check hasil push, bila berhasil maka akan muncul project yang kita push**



Sekian tutorial kali ini, sampai ketemu di tutorial selanjutnya. Happy learning happy sharing !!!